DEIM2022 H24-4

Transformerを用いた文書の自動品質評価

吉越　玲士†　　酒井　哲也†

† 早稲田大学基幹理工学部情報理工学科　〒 169-8555 東京都新宿区大久保 3-4-1
E-mail: †yoshikoshi@akane.waseda.jp, ††tetsuyasakai@acm.org

あらまし　一般に，自分が作成した文書の品質を客観的に評価するのは容易ではない．自己評価では，自分の思い入れが反映されてしまったり，自分では考えが及ばない観点からの評価ができないなどの問題があるためである．もし自分が作成した文書の品質の客観的な評価値が得られるのであれば，より良い文書作成の手助けになると考えられる．そこで本研究では，文書を適切な評価尺度で測ることができるユーザ支援システムとして，与えられた文書の品質を自動で定量的に評価する手法について検討する．そのためのアプローチとして，Transformer による文書の埋め込み表現を導入した．また，これに加えて，文書の品質にある程度相関のある複数の指標も併せて導入した．ここでのシステムによる文書品質評価の正確さは，あらかじめ与えられた手動による文書の評価値と予測値との平均二乗誤差および相関係数によって評価を行った．その結果，BERT による埋め込み表現に対して文書全体のトークン長とタグの文字数を与えた 770 次元の埋め込み表現を用いた機械学習において，文書評価値の予測システムは最も高い性能を発揮した．

キーワード　Transformer, BERT, monoBERT, Sentence–BERT, 自然言語処理, Wikipedia

1 はじめに

近年，インターネット上を通じてユーザが閲覧することができる文書の数は爆発的に増加した．消費者庁の令和３年版消費者白書 [1]によると，2021 年は「新しい生活様式」が浸透し，「インターネットの利用に費やす時間が増えた」と回答した人の割合が最も増えたという．また，インターネット上で利用しているものは情報収集（検索・閲覧）が約 9 割と割合が最も高く，消費者にとって最も重要な消費行動の一つだと言える．さらに，情報収集を行う動機となるオンライン学習の利用頻度も大きく増加しており，検索エンジンを通じて消費者が要求する文書には，年々より高い品質が求められつつある．

一般的に文書の品質を評価するためには，文書の読みやすさ，テキストの文体・文法，専門知識の量などの多岐にわたる要素を適切に吟味する必要がある [1]．そのため，高度な知識を有した者による評価でないと適切な評価尺度による評価が得られることは難しい．また，より良い品質の文書を記述するためには，現在の文書の状態を的確に理解している必要がある．文書の品質が適切に評価されずに過大評価してしまうと低品質の文書のままで満足してしまい，本来執筆することができたはずの品質水準まで達する機会を失いかねない．これは文書制作過程の中で一つの大きな難所となる．上記のような問題を解決するために，文書の品質を適切な評価尺度で測ることができるユーザ支援システムは一定の需要がある．

Shen ら [1] が行ったレンダリング画像を併用したマルチモーダルな文書の品質評価では，視覚的な埋め込み表現の取得に文書のスクリーンショット画像を使用している．そのため，掲載されているサイトのプラットフォームが変化するなどといったフォーマットの大きな変化を伴うと，画像を表現する埋め込み表現が大きく変化するため，適切な評価が困難になることが予想される．また，Shen ら [1] による実験では，テキストデータ以外の部分の暗黙的な部分にも品質を評価するための特徴があるとされる．

そこで本研究では，任意の文書から取得することが容易なテキストデータを主軸として文書の品質を評価することを試みた．テキストデータのみを使って評価することができれば，より広いユースケースでの活用ができ，様々な場面において文書の品質評価が可能になると想定されるためである．暗黙的な特徴を今回のテキストデータにおけるトークンの数や Markdown 記法のタグの文字数などで補完することができるのではないかと考え，実験を行った．

その結果，テキストデータにおけるトークンの数とタグの文字数の 2 種類の付加情報を monoBERT と併用した機械学習を用いる評価予測手法が最も高い性能を発揮した．よって，テキストデータの埋め込み表現のみでは捉えることのできない暗黙的な情報をこれら 2 種類の付加情報によって補完することができたと考えられる．

2 関連研究

2.1 レンダリング画像を併用した評価

文書の品質を評価するための情報は，テキスト自体に含まれているのみならず，添付されている画像などにも含まれていると考えるのが自然である．文書に対してより関連度の高い画像が添付されている方が，また，文書のテーマに関してより深い専門知識で言及されている方が，いずれの場合も一般的には文書の品質が高くなることが予想される．さらに，それ以外にも複数の要素が文書の品質に影響を与えている．

Shen ら [1] は，文書の品質には文書中に掲載されるテキスト

1：https://www.caa.go.jp/policies/policy/consumer_research/white_paper/2021/white_paper_127.html

データ以外の要素，例えばフォントの種類，画像，レイアウトなどの暗黙的な要素が文書品質へ影響を与えているとし，レンダリング画像を取り入れた文書の品質評価実験を行った．Shen らの実験では，文書中のテキストを双方向 LSTM [2] へ，文書のレンダリング画像を `Inception-V3` [3] にそれぞれ与え，それらの出力をフィードフォワード型のニューラルネットワークを用いて統合された埋め込み表現を生成し，品質評価予測を行った．

Shen らが行った品質評価実験では，学術論文データと Wikipedia の 2 つのドメインにおけるデータセットで彼らの手法が SOTA (state-of-the-art) の性能を発揮することができた．レンダリング画像から生成された埋め込み表現には，テキストデータからは表現することができない情報が含まれていることを追加実験で確認し，テキストデータとの相補性も確認された．

しかしながら，レンダリング画像を併用しているこのシステムは，学習に使ったデータセットへのみ最適化されている．そのため，文書が掲載されるウェブページ上の見た目が大きく変化してしまったり，異なるウェブページ上の文書であったりすると，埋め込み表現が大きく変化してしまうことが懸念される．前述のように，システムからの品質評価の予測にはレンダリング画像からの埋め込み表現の結果を用いているため，学習データセットへと特化しすぎた結果，汎化能力が失われてしまうことが予想される．

上記のような現象を避けるため，本研究ではレンダリング画像を用いない手法について比較検討を行う．

2.2 文脈一貫性を吟味した評価

文書の品質を評価するためには，テキストの意味だけでなく文脈一貫性も重要な要素になる．文脈的に一貫性の高い文書は可読性が向上するのはもちろんのこと，読者にとっての読む動機にもなり得るため，文書を執筆するユーザにとっては大変重要な要素である．また，文脈一貫性の低い文書は読者の誤解の誘因にもなる．

Liao ら [4] は，文書の品質には文脈が一貫して記述されているかどうかが大きな影響を及ぼしているとして，文脈がどの程度一貫したものであるかどうかを加味した評価モデル `HierCoh` を提案した．また，Liao らの行った実験では，文書を構成するそれぞれの文を Transformer [5] を用いて埋め込み表現とし，その結果を使ってテキストの一貫性をベクトルで取得した．

Liao らが行った小論文の品質評価実験では，Tagnipour と Ng による CNN+LSTM [6]，Dong らによる LSTM-CNN-att [7]，Tay らによる SkipFlow [8] を多くの場合で上回る結果を発揮した．テキストデータから文脈一貫性のデータを予測して品質評価を行う手法が有効であることが確認された．

しかしながら，文脈一貫性のデータを併用しているこのモデルは，図表といったデータを文書の評価に反映させることができない．前述のように，文書の品質は図表などテキストデータ以外にも含まれているため，それらを含めない限りは品質の評価値に曖昧さが発生することが予想される．そのため，本研究のデータセットとして用いる Wikipedia や Web ページのように図表を含む文書に関しては，文脈一貫性以外の複数の要素を用いなければ不適合な結果になることが考えられる．

上記のような理由から，本研究では Liao らと同様に Transformer を用いた手法について比較検討を行う．

3 実験手法

3.1 Transformer

自然言語処理に長けたモデルとして，Attention を中心に導入した Transformer [5] がある．Transformer は動作する際の処理として，まず文章を分解可能な最小単位であるトークンに分解する．分解されたトークン列中のある単語の埋め込み表現を得ようとしたときに，与えたトークン列の全トークンに対して Attention を向ける．あるトークン列全体の埋め込み表現を得ようとしたときの計算量は，トークン列長を n とすると $O(n^2)$ となる [9]．今回用いるデータセットのトークン列長の分布が図 1 のように中央値が約 2,000，最大値が約 40,000 と Transformer の一般的な入力トークン列長の制限 512 を大きく超えている．そのため，計算対象となるトークン列全てを用いて計算するのは非現実的になる．そこで，実験ではトークン列長を削減する前処理を導入した．

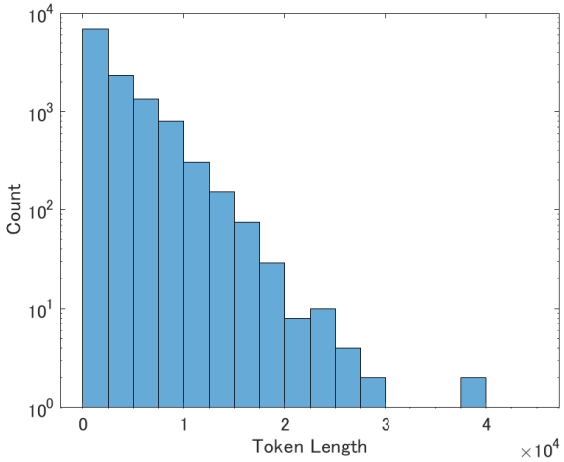

図 1　実験で使った文書のトークン長分布

実験では，文章の埋め込み表現を取得するために，Transformer として SOTA を達成した BERT [10] を利用した．具体的には，BERT を使って文章全体の埋め込み表現を取得することができる monoBERT と呼ばれる仕組みを導入した．この monoBERT は，トークン列の先頭に特殊トークン “[CLS]” を挿入し，このトークンに対する埋め込み表現を入力トークン列全体の埋め込み表現とする手法である．埋め込み表現の計算時には，先頭の特殊トークン “[CLS]” から全てのトークンに向けて Attention が向いているため，このトークンの埋め込み表現はテキスト全体を表現している埋め込み表現として扱うことができる [11]．

上記に加えて monoBERT と比較できる対象として，通常の BERT に対してプーリング層を導入した Sentence–BERT [12] を導入した．Sentence–BERT は monoBERT とは異なり，事前学習済みの BERT に対してプーリング層を導入して，テキスト全

体の埋め込み表現を得る仕組みを採用している．また，もう一つの大きな特徴としては計算時間を大きく短縮することができる点である．通常の BERT を使った実装ではタスクの実行に約 65 時間かかるような類似文書の検索タスクを，Sentence–BERT は 5 秒程度で実行することができる．

本研究では，以上の monoBERT と Sentence–BERT の 2 種類を主軸として実験を行い，その結果を比較検討する．

3.2 データセット

実験では，TREC 2021 Fair Ranking Track [13] で公開されているデータセットを用いた [2]．このデータセットには Wikipedia 上で掲載されている記事のうち，WikiProject[3] と呼ばれる，記事の品質を向上させることを目的としたプロジェクトによって文書の品質をスコア付けされたもの [4] を用いた．手動で文書に付された全てのスコアはそれぞれ 0 以上 1 以下の値で表現されていて，値の程度によって品質の高い順に以下の 6 つに分類されている．

- FA（Featured Article）
- GA（Good Article）
- B–class
- C–class
- Start
- Stub

実験では，各クラスに属する記事の数を 2,000 件，全体で 12,000 件となるようにデータを選別し，その上で学習，検証，テストを行った．使用したデータに含まれる文書の品質スコアは図 2 のような分布となった．この際学習には全体の 70%，検証とテストにはそれぞれ全体の 15%のデータを事前にシャッフルした上で重複しないように割り当てた．後述するサンプル数の異なる実験でも，同様の割合，同様の手法でデータを選別した．

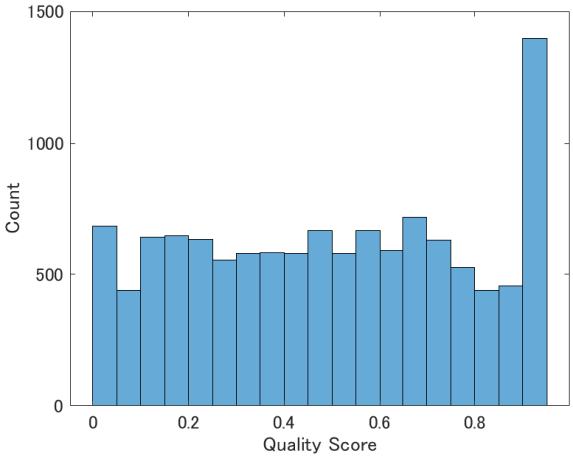

図 2　実験で使った文書の品質評価スコア分布

2：https://fair-trec.github.io/2021/

3：https://en.wikipedia.org/wiki/Wikipedia:WikiProject

4：https://en.wikipedia.org/wiki/Wikipedia:Content_assessment

3.3 データの前処理方法

今回使用したデータセット [13] では，Wikipedia に含まれる Markdown がオリジナルの状態で含まれている．この Markdown の部分に関しては，例えば実際に Wikipedia 上に含まれるカテゴリを示すための表記

`[[Category: Southern United States]]`

をトークン化すると，

`"[CLS]", "[", "[", "category", ":", "southern", "united", "states", "]", "]", "[SEP]"`

と 11 のトークンに分解されるように，入力トークン列長を可能な限り少なくさせる本実験の方針と逆行するような結果となる．そこでこの問題を解決するために，Wikipedia 内で使用される Markdown の代表的な記法については削除するような加工を施し，可能な限りテキストデータのみが含まれるようにする加工を行った．加工では，Wikipedia のテンプレートやテーブル，画像挿入用の HTML タグを表現する表 1 に挙げる全ての Markdown は開始部分のタグから終了部分のタグまで全体をスクリプトで検出したり正規表現を用いたりして削除した．

表 1　削除した Markdown とその機能

方法	Markdown の表記	機能
†	{{...}}	各種テンプレート
	{|...|}	表
††	/<ref.*?>.*?</ref>//	脚注・注釈・出典情報の挿入
	/<ref.*?/>//	脚注・注釈・出典情報の挿入
	/={1,3}.*?={1,3}//	見出し
	/\[\[Category:.*?\]\]//	カテゴリ情報
	/\'{1,3}(.*?)\'{1,3}/$1/	強調（斜体か太字）

† はプログラムで対になるものを検出し，開始から終了までを削除した．

†† は正規表現を使ったため，“/検索文字列/置換先文字列/” という記法で示している．

この加工の後に，各種 Transformer にテキストデータをトークン化した上で入力する．しかし，実際の計算ではトークン化にも埋め込み表現の計算にも多くの時間を要する．そのため，計算時間を短縮させるための一環として，5,000 字以上の入力テキストに関しては，それ以下の文字数になるようにあらかじめ切り捨てた上でトークン化を行った．最終的に，各種 Transformer によって計算された文書に含まれるテキストデータの埋め込み表現を E とする．

前述の通り，文書の品質は今回削除する部分なった図表に相当する Markdown にも依存するため，単に情報を削除してしまうと品質評価の予測精度が悪化することが予想される．よって，ここでは何も加工していない状態の文書から直接計算したトークン列長 T と，Markdown を削除する前後の文字数の差，つまり Markdown 部分のみの文字数 M を別途計測し，それらを埋め込み表現の末尾に付加情報として含めた場合についても，どのように予測精度に影響を及ぼすのかを調べた．

3.4 用意した実験条件

表 2 の (A)〜(E) の 5 つの条件でそれぞれ実験を行った．付

加情報のトークン列長 T と Markdown 部分文字数 M は，各種 Transformer から出力される 768 次元の埋め込み表現ベクトルの出力 E の末尾にそれぞれ付けるように追加したため，例えば (B) の最終的な文書の埋め込み表現は $[E\ T\ M]$[5]のように 770 次元となっている．

表 2　実験条件ごとの埋め込み表現の計算方法一覧

	Transformer の種類	付加情報	次元数
(A)	monoBERT	なし	768
(B)	monoBERT	T と M	770
(C)	monoBERT	T	769
(D)	monoBERT	M	769
(E)	Sentence–BERT	なし	768

また，Sentence–BERT の事前学習済みモデルとして多数の選択肢が存在するが [6]，実験では `distilroberta-base`[7]をベースとして作成された `all-distilroberta-v1`[8]を用いた．この事前学習済みモデルは，様々なユースケースに対応するようにチューニングされていて，Reddit に投稿されたコメントをはじめとした 10 億以上の文の組が含まれた学習データセットを使ってトレーニングされたものである．入力トークン列の最大長はデフォルトでは学習時と同じく 128 に制限されていて，出力が今回使用した BERT [10] と同じく 768 次元のベクトルとなっている．出力前のプーリングについては複数種類の中から選ぶことができるが，今回はその中でも最も性能が高い Average Pooling を選択した．

768〜770 次元の埋め込み表現 (A)〜(E) を生成した後に，ニューラルネットワークを用いて教師あり学習を行う．ニューラルネットワークでの学習では，学習アルゴリズムを Levenberg–Marquardt 法 (LM 法) に設定した．LM 法は最急降下法やガウス・ニュートン法などと同様に，目的関数の勾配を利用してネットワークの最適化を行う．この学習アルゴリズムは，MATLAB で利用可能な他の 2 つのアルゴリズム，ベイズ正則化 (Bayesian regularization backpropagation) やスケーリング共役勾配法 (scaled conjugate gradient backpropagation) に比べると，メモリをより多く消費する代わりに実行時間が比較的高速であるのが特徴である．

本実験では，LM 法のパラメータとしてニューラルネットワークの隠れ層 (hidden layer) の数 H がどのように結果に影響を及ぼすのかを併せて検証するために，隠れ層数 H を 5，10，15，20 の 4 つの場合について検討した．さらに，どの程度のサンプル数 N が最適であるのかも併せて検証するために，全体のサンプル数 N を 2,000，6,000，12,000 の 3 種類で行った．これらのサンプルを学習：検証：テストを 70%：15%：15%の割合で分割することで実験を行った．

以上の条件をまとめると表 3 のようになる．実験は，これらを全て網羅するように行った．

表 3　実験で比較対象としたモデル，学習条件一覧

	monoBERT	Sentence–BERT
Transformer について		
使用モデル名	base	all–distilroberta–v1
プーリング層	なし	Average
ニューラルネットワークについて		
学習アルゴリズム	Levenberg–Marquardt 法	
隠れ層の数 H	5〜20 の 4 種類	
サンプルの数 N	2,000〜12,000 の 3 種類	

最終的にニューラルネットワークから出力される値は，文書の品質評価スコアの予測値となる．ただし，ネットワークの出力値は今回の評価値の範囲である 0 以上 1 以下の範囲に収まる保証はないため，評価値としてもあり得ない値を出力する可能性がある．そこで，テストケースの評価予測値に限っては 0 未満のものは 0 に，1 より大きいものは 1 に揃えた上で評価を行った．この予測値の評価は，使用したデータセットに含まれているあらかじめ手動でラベル付けされているテストケースに付されている品質評価スコアとの平均二乗誤差 *MSE* (Mean Squared Error) と相関係数 R で行った．

MSE は予測値と正解値との差の二乗を平均することで求めることができ，しばしば機械学習の損失関数に用いられる．全てのテストケースで正解値を出すことができる理想的なシステムに近いほど *MSE* は 0 へと近づく．相関係数 R は，予測値と正解値との線形関係の程度を表す指標で，−1 以上 1 以下の値をとる．今回の場合，理想的なシステムに近いほど正の相関があるため R は 1 へと近づく．

4　実験結果

4.1　評価実験

実験結果は，*MSE* が表 4，相関係数 R が表 5 のようにそれぞれなった．それぞれの表中で最も優れている値には下線を付してある．

埋め込み表現 (A)〜(E) で比較すると，*MSE* の場合でも R の場合でも，(B) が最も優れていた．また，monoBERT の埋め込み表現のみの (A) と Sentence–BERT の埋め込み表現のみの (E) を比較したときでは，ほぼ同水準となった．

学習条件中で比較すると，サンプル数 N は多い方が，隠れ層の数 H は少ない方が，それぞれ結果が優れている傾向となった．

埋め込み表現の生成方法で見ると，monoBERT から出力された埋め込み表現に対して付加情報 T と M を与えたときの条件 (B) のときに，最も高い性能を記録した．さらに条件 (B) では，サンプル数 N = 12,000 と隠れ層の数 H = 5 のときに，*MSE*= 0.0036，R = 0.9772 といずれの評価においても，全ての実験結果の中で最も高い性能を記録した．

また，今回最も高い性能を発揮した条件 (B)，条件 (A) および条件 (E) の N = 12,000 と H = 15 であった際の，それぞれのテストケースにおける予測値と正解値の分布は図 3，図 5 および図 7 のようになった．条件 (A) と (E) は，条件 (B) に対するベースラインとなっている．今回，表 5 で示している値は，そ

5：MATLAB におけるベクトルの結合記法.

6：https://www.sbert.net/docs/pretrained_models.html

7：https://huggingface.co/distilroberta-base

8：https://huggingface.co/sentence-transformers/all-distilroberta-v1

表 4　各種条件下におけるスコア予測値の平均二乗誤差 MSE ($\times 10^{-4}$)

学習条件		(A)	(B)	(C)	(D)	(E)
$N = 2000$	$H = 5$	455	179	224	199	479
$N = 2000$	$H = 10$	386	163	318	272	444
$N = 2000$	$H = 15$	501	231	353	287	466
$N = 2000$	$H = 20$	557	188	399	396	515
$N = 6000$	$H = 5$	360	42	166	103	363
$N = 6000$	$H = 10$	303	53	211	153	486
$N = 6000$	$H = 15$	295	104	213	162	361
$N = 6000$	$H = 20$	452	120	278	176	353
$N = 12000$	$H = 5$	348	36	145	77	321
$N = 12000$	$H = 10$	297	45	193	93	320
$N = 12000$	$H = 15$	339	37	193	179	364
$N = 12000$	$H = 20$	314	96	215	200	369

（最小値は下線で示す.）

表 5　各種条件下におけるスコア予測値の相関係数 R ($\times 10^{-4}$)

学習条件		(A)	(B)	(C)	(D)	(E)
$N = 2000$	$H = 5$	7584	8827	8525	8599	6788
$N = 2000$	$H = 10$	7532	8856	7826	8307	7215
$N = 2000$	$H = 15$	6760	8492	7781	8281	6830
$N = 2000$	$H = 20$	5817	8790	7840	7372	6150
$N = 6000$	$H = 5$	7433	9758	8875	9382	7445
$N = 6000$	$H = 10$	7967	9686	8596	8999	7033
$N = 6000$	$H = 15$	7977	9351	8550	8987	7551
$N = 6000$	$H = 20$	7250	9280	8211	8859	7736
$N = 12000$	$H = 5$	7624	9772	9070	9507	7882
$N = 12000$	$H = 10$	7977	9709	8863	9383	7840
$N = 12000$	$H = 15$	7835	9765	8832	8900	7497
$N = 12000$	$H = 20$	7922	9401	8613	8780	7449

（最大値は下線で示す.）

れぞれのグラフ上に示した相関係数 R の値に対応する．いずれの場合も，理想的なシステム，つまり予測値と正解値が完全に一致する場合では散布図の点線上にプロットが表示される．

前述の図 3，図 5 および図 7 のそれぞれに示す条件での機械学習において，学習用データに対する予測値と実際の値の分布は図 4，図 6 および図 8 のようにそれぞれなった．

また，図 3，図 5 および図 7 についての混同行列は図 9，図 10 および図 11 のようにそれぞれなった．

図 3 と図 4 では，学習データセットとテストケースのいずれに対しても相関係数 R が 0.97 以上と高い値で収束した．一方で，図 6 と図 8 では，学習時の散布図と相関係数 R から見るとある程度の精度で品質を評価することができている．しかし，いずれの場合でもテストケースである図 5 と図 7 での結果では大きく予測値が正解値と外れるパターンも存在する．相関係数 R も学習データのものと比べると 0.13〜0.07 ほどテストケースで値を下げているため，過学習 (overfitting) が発生していると考えられる．上記の過学習の傾向は，テキストデータの埋め込み表現のみで予測している仕組み自体に原因があると予想される．また，よりテストケースでの性能低下が著しい条件 (E) については，Sentence–BERT の入力可能なトークン数が通常の BERT に比べて少ないことも要因の一つとして挙げられる．条件 (A) と (E) を踏まえた上で条件 (B) の結果を評価すると，テキストデータの埋め込み表現では捉えることができない暗黙的な情報をトークン列長とタグの文字数の 2 つで補完することができていると考えられる．

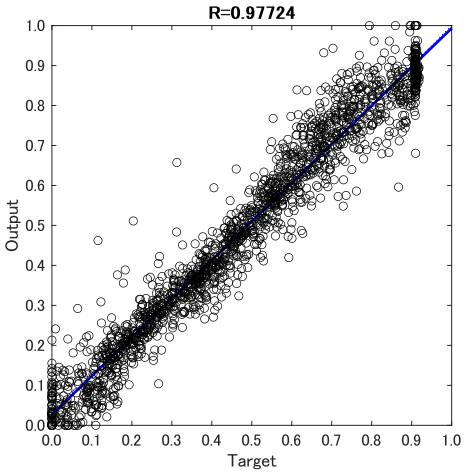

図 3　条件 (B) のサンプル数 $N = 12{,}000$，隠れ層の数 $H = 5$ であったときのテストケースに対するシステムの予測値と正解値との相関（各グラフの丸プロットがモデルによる予測値，実線が予測値の線形回帰．モデルの予測値が正解値だと点線上にプロットされる．以下同じ.）

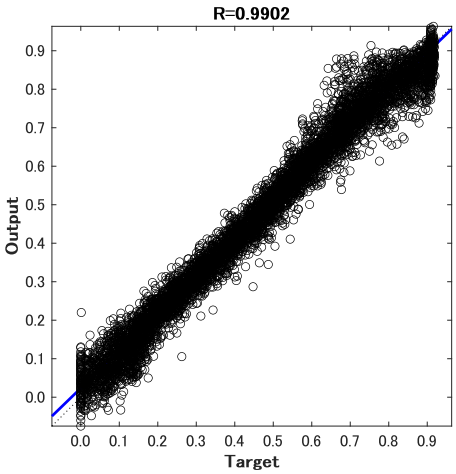

図 4　条件 (B) のサンプル数 $N = 12{,}000$，隠れ層の数 $H = 5$ であったときの学習データに対するシステムの予測値と正解値との相関

4.2　失敗分析

条件 (A) の下で行った実験について，予測値と正解値が著しく離れている場合のテストケースを抽出し，その原因について考える．ニューラルネットワークの学習条件は，条件 (B) において最も高い性能を発揮した隠れ層数 $H = 15$，サンプル数 $N = 12{,}000$ に揃えた上で分析を行った．すると，以下に挙げるような特徴を持つ文書では，予測値が大きく外れることが分かった．

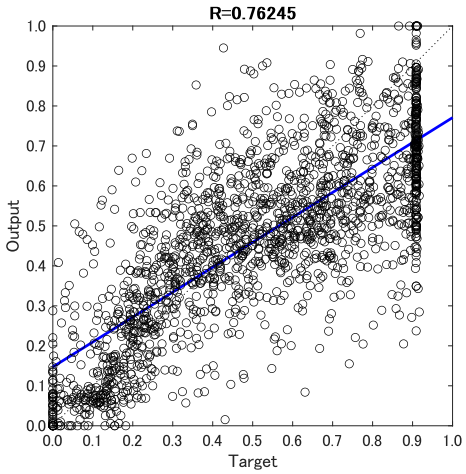

図 5　条件 (A) のサンプル数 N = 12,000，隠れ層の数 H = 5 であったときのテストケースに対するシステムの予測値と正解値との相関

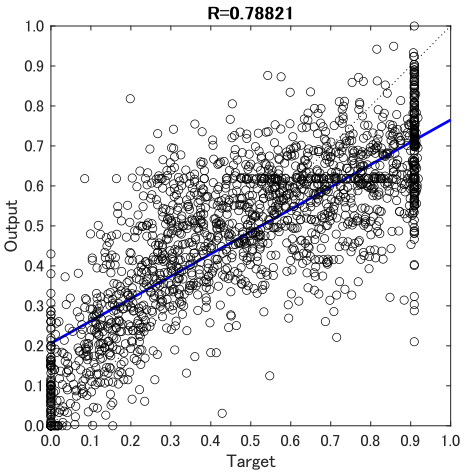

図 7　条件 (E) のサンプル数 N = 12,000，隠れ層の数 H = 5 であったときのテストケースに対するシステムの予測値と正解値との相関

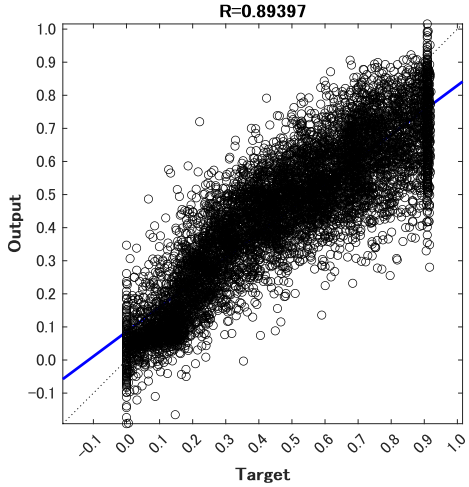

図 6　条件 (A) のサンプル数 N = 12,000，隠れ層の数 H = 5 であったときの学習データに対するシステムの予測値と正解値との相関

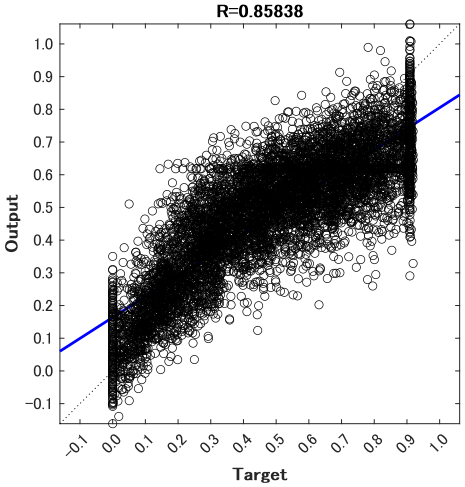

図 8　条件 (E) のサンプル数 N = 12,000，隠れ層の数 H = 5 であったときの学習データに対するシステムの予測値と正解値との相関

(1) Markdown の部分が占める割合が大きい

文書に含まれる内容に Markdown が占める割合が著しく大きい場合は，Transformer に入力する前処理としてそれらを可能な限り削除しているため，入力トークン列が元の文書に比べて大きく減ることがある．このような場合だと，本来であれば評価するべき箇所である図表などに対しての情報量は一切除かれるため，テキストデータ自体は品質の低い文書に似たものになることがある．

(2) 文書に含まれるテキストデータが極端に小さい

文書の品質を分類する 6 つのクラスのうち，最も低い Stub に分類される文書の場合，内容が箇条書きや他の文書への参照情報で完結しているものが多い．

例えば，日本語版 Wikipedia の「曖昧さ回避」[9]に相当する文書になるため，他の文書への参照情報のみの記載で完結している．このような文書に対して Markdown の削除を行うと，最終的に Transformer へ入力されるテキストデータのトークン列がほとんど残らない状態となる．本実験では，品質の極端に低い文書も学習データに含めるように加工しているため，これは学習データ数自体の不足ではなく，予測するための特徴が少なすぎて誤った評価を行っている結果だと予想される．

(3) 文書に含まれるテキストデータが極端に大きい

Transformer に入力するテキストデータは，計算時間の観点から予めある程度の長さで切り落とした上で入力される．ここで極端に文書のボリュームが大きい場合，序盤のみを加味した Transformer による埋め込み表現を使って予測することになる．そのため，その先にどの程度文章が続くのかという付加情報が存在しない状態では，学習の時点でも予測する時点でも困難が生じる．

9：https://ja.wikipedia.org/wiki/Wikipedia:曖昧さ回避

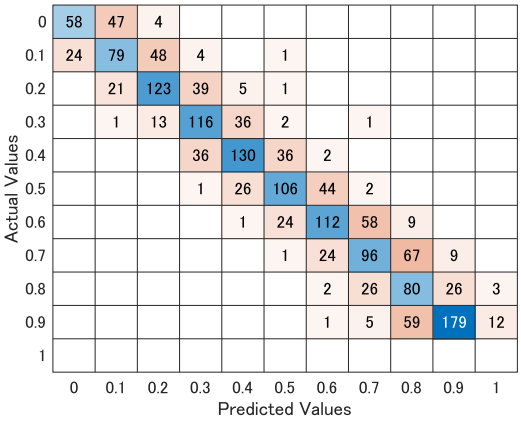

図 9　条件 (B) のサンプル数 $N = 12{,}000$，隠れ層の数 $H = 5$ であったときのテストケースに対するシステムの予測値と正解値の混同行列（対角要素が予測値と正解値が一致した場合．以下同じ．）

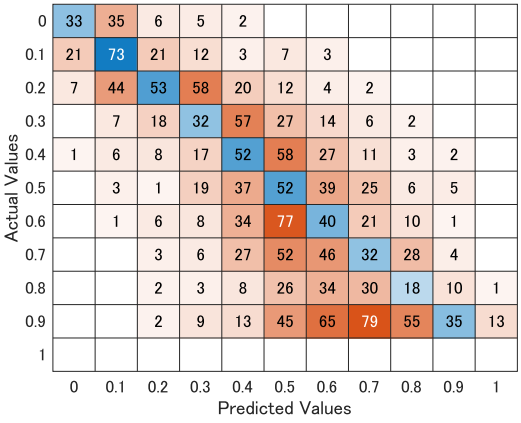

図 10　条件 (A) のサンプル数 $N = 12{,}000$，隠れ層の数 $H = 5$ であったときのテストケースに対するシステムの予測値と正解値の混同行列

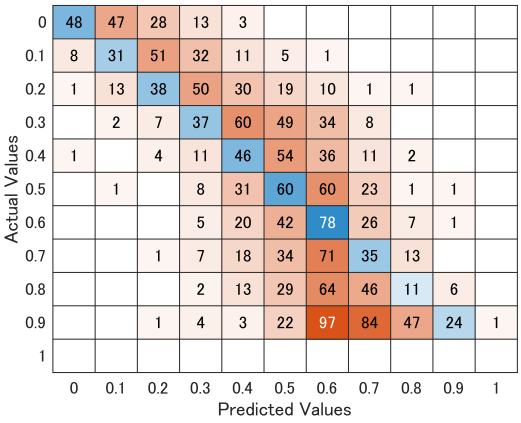

図 11　条件 (E) のサンプル数 $N = 12{,}000$，隠れ層の数 $H = 5$ であったときのテストケースに対するシステムの予測値と正解値の混同行列

5　結論・今後の課題

5.1　ま　と　め

本研究では，テキストデータのみを用いたシステムで，より幅広い場面において活用が見込まれる文書の品質評価の予測システムを提案した．テキストデータのみを用いることで予測することができれば，形態や場所を選ばずに活用することができるため有用なシステムとなることが予想される．

実験では 5 つの埋め込み表現の計算方法を比較したが，その内の monoBERT の出力に対して入力トークン長とタグの文字数を付した条件 (B) が最も優れていた．また，BERT の埋め込み表現を直接用いる条件 (A) と，同様の条件下で Sentence–BERT の出力を用いた条件 (E) を比較した場合だと，条件 (E) の方が劣る結果となった．条件 (A) と (E) は，いずれの場合もテストデータにおける予測性能が学習データにおけるそれを下回っているため，過学習の傾向がある．さらに条件 (A) と (E) の間の結果の差を見てみると，条件 (E) の方が過学習の傾向がより顕著に見られた．上記の原因としては，Sentence–BERT の入力可能トークン数が BERT よりも高速化の観点から大きく削減されている点が挙げられる．

5.2　今後の課題

今回の実験では Wikipedia のデータのみを用いているため，適用可能な文書の範囲が目的に適わないことが予想される．しかしながら，テキストデータの埋め込み表現に対してトークン列長とタグの文字数を付加することで，ある程度テキストデータに存在しない暗黙的な情報を補完することができた．より汎用的な予測システムを実用化させるためには，一般的には様々な文書の学習データが必要になる．このため，幅広い文書に対応できるように学習セットを豊富に用意した上で，再び同様の実験を行うことが課題となる．

文　　献

[1] Aili Shen, Bahar Salehi, Jianzhong Qi, and Timothy Baldwin. A general approach to multimodal document quality assessment. 68:607–632.

[2] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision.

[4] Dongliang Liao, Jin Xu, Gongfu Li, and Yiru Wang. Hierarchical coherence modeling for document quality assessment. 35(15):13353–13361.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.

[6] Kaveh Taghipour and Hwee Tou Ng. A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891. Association for Computational Linguistics.

[7] Fei Dong, Yue Zhang, and Jie Yang. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162. Association for Computational Linguistics.

[8] Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. SkipFlow: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring.

[9] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding.

[11] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond.

[12] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks.

[13] Michael D. Ekstrand, Graham McDonald, Amifa Raj, and Isaac Johnson. Overview of the TREC 2021 fair ranking track.